Hosts anlegen

Im ersten Schritt werden aus allen WatResponses alle eindeutigen Hostnames gemeinsam mit den IP-Adressen, die sie verwenden, herausgefiltert und als "Host"-Nodes in Neo4J angelegt.

```
MATCH (resposne: WatResponse)
WITH DISTINCT response.host AS hostname, COLLECT(DISTINCT
response.ip_address) AS host_ips
CREATE (h: Host { name: hostname, ips: host_ips })

+-------------------+
| No data returned. |
+-------------------+
Nodes created: 32857
~34654 ms (average of 10 runs)
```

Links anlegen

Im zweiten Schritt werden alle Link-Relationships zwischen den einzelnen Hosts angelegt. Dabei wird darauf geachtet, dass kein Link doppelt angelegt wird, dies geschieht automatisch mit dem "MERGE"-Befehl. Dieser überprüft ob eine angegebene Node oder Relationship zwischen Nodes bereits existiert und legt sie an falls es nicht gibt.

```
MATCH (response: WatResponse)
WITH response.host AS hostname, COLLECT(DISTINCT response.linked_hosts) AS
linked_hosts
UNWIND linked_hosts AS linkedhosts
WITH hostname, FILTER(x IN linkedhosts WHERE NOT x = hostname) AS
unique_links
UNWIND unique_links AS target
MERGE (h1: Host {name: hostname})
MERGE (h2: Host {name: target})
MERGE (h1)-[:Link]->(h2);

+-------------------+
| No data returned. |
+-------------------+
Nodes created: 18172
Relationships created: 40714
Properties set: 18172
Labels added: 18172
~16486 ms (average of 10 runs)
```

Im ersten Anlauf wurde versucht, die Links ohne Index anzulegen. Da dies nach 15 Minuten immer noch nicht abgeschlossen war, wurde der Befehl abgebrochen und zuvor ein Index angelegt.

Index anlegen

```
CREATE INDEX ON :Host(name)
```

Damit wird ein Index auf das „name“-Attribut der Host-Nodes angelegt. Dadurch kann das Anlegen der Link-Relationships deutlich performanter durchgefürt werden. Alle ~40.000 Relationships werden durchschnittlich in etwas mehr als 15 Sekunden angelegt (siehe oben).

Vergleich Index <--> kein Index

Zum Vergleich wurde vor Anlegen des Index für eine einzelne Seite, die auf 8 andere Seiten verlinkt, getestet, wie sich der Befehl mit und ohne Index unterscheidet. Das Anlegen dieser 8 Relationships dauerte durchschnittlich ohne Index 7731ms, mit Index hingegen nur 543ms. In diesem Fall unterscheidet sich die Performance dieses Befehls im Vergleich kein Index / Index um etwa 142%.

Eindeutigkeit von Host-Nodes und Links

Die folgenden beiden Befehle zeigen, dass sowohl alle Host-Nodes als auch alle Links zwischen zwei Hosts in der Neo4J-Datenbank eindeutig sind.

```
MATCH(response: Host)
WITH response.name AS host, COUNT(response.name) as cnt
WHERE cnt > 1
RETURN host;

+------+
| host |
+------+
+------+
0 row
825 ms
```

```
MATCH (h1: Host)-[l: Link]->(h2: Host)
WITH h1, h2, COUNT(l) AS link_count
WHERE link_count > 1
RETURN h1, h2, link_count;

+----------------------+
| h1 | h2 | link_count |
+----------------------+
+----------------------+
0 row
794 ms
```

Datenmengen

Mit den untenstehenden Befehlen werden die Datenmengen dokumentiert, die durch die WatResponses angelegt wurden.

Anzahl der Nodes

```
MATCH ()
RETURN COUNT(*) AS node_count;

+------------+
| node_count |
+------------+
| 249875     |
+------------+
1 row
770 ms
```

Anzahl der Relationships

```
MATCH ()-->()
RETURN COUNT(*) AS rel_count;

+-----------+
| rel_count |
+-----------+
| 239550    |
+-----------+
1 row
53 ms
```

Fragen an die Karte

Auf welche Hosts wird am öftesten verlinkt? (Top 10)

Als erste Frage an die Karte wollten wir herausfinden, auf welche Hosts am öftesten verlinkt wird. Die Sieger sind, wenig überraschend, die großen Social-Media-Seiten Facebook und Twitter. Da wir nur ein sehr begrenztes Datenset des Commoncrawl-Projekts verwenden, sind diese Zahlen natürlich alles andere als exakt.

```
MATCH (h: Host)<-[l: Link]-()
RETURN h.name AS hostname, COUNT(l) AS cnt
ORDER BY cnt DESC
LIMIT 10;

+----------------------------------+
| hostname                  | cnt  |
+----------------------------------+
| "www.facebook.com"        | 2033 |
| "twitter.com"             | 1322 |
| "b.scorecardresearch.com" | 752  |
| "www.youtube.com"         | 562  |
| "plus.google.com"         | 543  |
| "www.blogger.com"         | 364  |
| "www.google.com"          | 313  |
| "www.twitter.com"         | 278  |
| "instagram.com"           | 253  |
| "2.bp.blogspot.com"       | 235  |
+----------------------------------+
10 rows
635 ms
```

Welche Hosts verwenden die meisten IP-Adressen? (Top 10)

Als nächstes wollten wir herausfinden, welche Hosts die meisten unterschiedlichen IP-Adressen verwendet. Zu diesem Zweck wurde einfach die Länge des IP-Adressen-Array ermittelt und das Ergebnis dann nach absteigenden Zahlen sortiert.

In unserem begrenzten Datenset verwendet Youtube 20 verschiedene IP-Adressen für ihren Hostnamen „www.youtube.com“.

```
MATCH (h: Host)
WHERE NOT h.ips IS NULL
RETURN h.name AS hostname, length(h.ips) AS cnt
ORDER BY cnt DESC
LIMIT 10;

+------------------------------+
| hostname                | cnt |
+------------------------------+
| "www.youtube.com"       | 20  |
| "www.pinterest.com"     | 18  |
| "www.google.com"        | 17  |
| "www.reddit.com"        | 15  |
| "www.rentalhouses.com"  | 15  |
| "www.popsugar.com"      | 14  |
| "www.cnet.com"          | 14  |
| "forecast.weather.gov"  | 14  |
| "www.gamezone.com"      | 13  |
| "www.terrysvillage.com" | 13  |
+------------------------------+
10 rows
473 ms
```

Wie viele Hosts verwenden ein Content Delivery Network?

Zum Abschluss wollten wir noch überprüfen, wie viele Hosts ein Content Delivery Network verwenden, um ihre Inhalte schneller an ihre Nutzer zu liefern.

Anmerkung: Hier wird nur geprüft, von wie vielen Hosts es ausgehende Links gibt, die "cdn" in ihrem Hostname haben. Theoretisch kann es natürlich auch Content Delivery Networks geben, die "cdn" nicht in ihrem Hostname haben.

```
MATCH (h: Host)-[l: Link]->(h2: Host)
WHERE h2.name CONTAINS 'cdn'
RETURN COUNT(h) AS count_cdn;

+-----------+
| count_cdn |
+-----------+
| 800       |
+-----------+
1 row
285 ms
```

```
MATCH (h: Host)-[l: Link]->(h2: Host)
WHERE NOT h2.name CONTAINS 'cdn'
RETURN COUNT(h) AS count_no_cdn;

+--------------+
| count_no_cdn |
+--------------+
| 39914        |
+--------------+
1 row
422 ms
```

Fazit: 800 Hosts verlinken auf ihrer Seite auf eine andere Seite, die "cdn" in deren Hostnamen hat. Im Gegensatz dazu gibt es 39.914 Hosts, die das nicht tun. Das ist somit ein Anteil von knapp 2%.

Nützliche Hilfsfunktionen

Löschen einer bestimmten Relation:

```
MATCH ()-[l: Link]->()
WHERE ID(l) = {ID of Relation to be deleted}
DELETE l;
```

Anzeige aller Hosts und deren verlinkter Hosts:

```
MATCH (h1: Host)-[l: Link]->(h2: Host)
RETURN h1, l, h2;
```